

# Symcloud

## Motivation

Filehosting-Plattformen sind in der heutigen Zeit allgegenwärtig. Ohne einen Zugriff zu einem der allgemein verfügbaren Dienste, ist heutzutage eine Zusammenarbeit in einer Gruppe von Menschen, zum Beispiel an einer Universität, kaum möglich.

In den letzten Jahren wurde allerdings, nicht zuletzt durch den Wistleblower Edward Snowden bekannt, dass genau diese Daten von verschiedensten Quellen abgesaugt und verarbeitet werden.

Diese Angst vor Kontrollverlust bewegt viele Menschen dazu, Ihre Daten am liebsten bei sich zu behalten und mit quelloffenen Lösungen die Daten auf Ihrem eigenen Server zu speichern.

Meine Masterarbeit beschäftigt sich mit der Konzeption einer Speicherlösung, die Ideen aus verschiedenen Applikationen und Technologien kombiniert und es damit den Anwendern ermöglichen soll, Ihre Daten lokal und sicher zu speichern. Diese aber bei Bedarf gezielt an andere Benutzer weiterzugeben, selbst wenn diese nicht auf demselben Server registriert sind.

## Agenda

Aber nun zur Agenda. Ich habe diese Präsentation in vier Teile aufgeteilt:

1. Eine Einführung in das Thema und die Inspirationsquellen
2. Die Resultate und das entstandene Konzept
3. Die Implementierung und die entstandenen Software Komponenten
4. Und am Ende noch kurz zu den Herausforderungen und ein kurzes Fazit als Abschluss diese Präsentation

## Einführung

Die Idee entstand vor ein paar Jahren, als ich auf ein Open-Source Projekt namens ownCloud gestoßen bin. Diese Idee reifte in den letzten Monaten zu meiner Masterarbeit heran.

OwnCloud ist eine freie Software für das Speichern von Daten auf einem eigenen Server. Bei Einsatz eines entsprechenden Clients wird dieser automatisch mit einem lokalen Verzeichnis synchronisiert. Über eine Plattform lassen sich die Daten auch im Browser bearbeiten und verwalten.

Zuerst ging es bei dem Projekt lediglich um eine Plattform die es ähnlich wie ownCloud ermöglichen soll, Dateien zu verwalten und mit anderen zu teilen. Bei der Recherche jedoch erkannte ich, dass die Unabhängigkeit von einem Provider sehr wichtig für die Zielgruppe des Projektes ist. Jeder der Anwender will sein eigener Provider sein. Jedoch ist es bei klassischen Systemen wie ownCloud oder auch kommerziellen Anbietern nur schwer möglich, Dateien über die Grenzen des eigenen Servers bzw. Anbieters zu teilen.

Daher wurde der Fokus zu Beginn der Projektzeit sehr schnell auf genau diesen Anwendungsfall gelegt.

Auf der Suche nach Referenzprojekten bin ich schnell auf zwei sehr starke Inspirationsquellen gestoßen. Beide Projekte sind in ihrer Zeit gesehen Pilotprojekte und Vorreiter ihrer Technologien.

## Projekt Xanadu

Das Projekt Xanadu wurde durch Theodor Holm Nelson - kurz Ted - in den 1960er Jahren initiiert und ist bis heute nicht finalisiert worden. Jedoch prägte Ted Nelson mit der Gründung des Projektes und durch die

Veröffentlichung des wissenschaftlichen Artikels “The Hypertext” im Jahre 1965 den Begriff des Hypertext und inspirierte damit unter anderem Tim Berners-Lee zu der Entwicklung des World-Wide-Web mehr als zwei Jahrzehnte später.

Die Kernaussage des Projektes beinhaltet 12 Thesen, die zum Teil im heutigen Web Anwendung finden. Einige dieser Thesen, die nicht oder nur zum Teil umgesetzt wurden, sind für symCloud sehr interessant und werden in der Arbeit genauer betrachtet.

## **Diaspora**

Diaspora ist ein dezentrales soziales Netzwerk. Es ist vergleichbar mit Facebook, welches jedoch zentralisiert aufgebaut ist. Das bedeutet, dass jeder Benutzer der mit anderen kommunizieren will, auch bei Facebook registriert sein muss.

Diaspora geht hier einen anderen Weg. Über ein spezielles Protokoll können z.B. Nachrichten auch mit Benutzern ausgetauscht werden, die auf einem anderen Diaspora Server registriert sind.

Diese Architektur erlaubt es den Anwendern, Ihre eigenen Netzwerke zu gestalten und Ihre Daten nur denjenigen zur Verfügung zu stellen, mit denen Sie sie teilen wollen.

## **Ziele**

Aus diesen Inspirationsquellen wurde das Ziel ausgegeben, ein funktionsfähiges und erweiterbares Konzept zu erstellen, mit dem die Vorteile dieser Projekte vereint werden können. Es soll dem Anwender die Möglichkeit geben, seine Daten privat auf seinem Server zu speichern und Sie denjenigen zur Verfügung stellen, mit denen Sie die Daten teilen wollen.

Als zusätzliches Ziel wurde ausgegeben, dass große Teile dieses Konzeptes in einem Prototypen realisiert werden, um die Funktionstüchtigkeit zu beweisen.

## **Anforderungen**

Aus den drei Inspirationsquellen wurden die Anforderungen an ein solches System formuliert und in drei Teilgebiete unterteilt.

1. Datensicherheit
2. Filehosting und -sharing Funktionalitäten
3. Architektur

Zusätzlich wurde darauf geachtet, dass der Prototyp auf dem aktuellen Stand der Technik und mit einem Fokus auf Wartbarkeit und Erweiterbarkeit umgesetzt wird.

## **Konzept**

Um ein ausgewogenes Konzept zu erstellen, wurden Technologien aus den Bereichen:

- Verteilte Datenmodelle - GIT
- Verteilte Daten - Diaspora
- Verteilte Dateisysteme - NFS und XtremFS
- Objekt Speicherdienste - Amazon S3
- Datenbankgestützte Dateiverwaltung - GridFS

evaluiert.

Die jeweiligen Vorteile dieser Systeme wurden in symCloud kombiniert, um nur ein paar Beispiele zu nennen:

- Das Datenmodell von GIT wurde erweitert, um die Anforderungen vollständig zu erfüllen.
- Die lose gekoppelte Server-Architektur von Diaspora findet auch Anwendung in der Architektur von symCloud.
- Mithilfe einer einfachen Variante des Primärbasierten-Backup-Protokolls von XtreamFS werden die Objekte in einem Netzwerk von Servern verteilt. Diese Verteilung wird unter anderem von den Benutzerrechten gesteuert. Dies bedeutet, dass diejenigen Server eine Kopie der Daten erhalten, auf denen mindestens ein Benutzer registriert ist, der Zugriff auf die Daten besitzt.

Durch die Kombination dieser Technologien konnte ein Konzept erstellt werden, das alle Anforderungen erfüllt und für Weiterentwicklungen genug Freiraum bereitstellt.

## Implementierung

Bei der Implementierung wurde auf bewährte Technologien aus dem Umfeld von PHP und dem Framework Symfony2 vertraut. Auf diese Technologien möchte ich hier nicht genauer eingehen. Dies würde den Rahmen dieser Präsentation sprengen.

Entstanden ist während der Projektzeit ein funktionstüchtiger Prototyp, der einen Großteil des Konzeptes implementiert.

Es wurde eine Bibliothek implementiert, dass eine einfache verteilte Datenbank enthält. Die Daten werden lokal in einem Ordner abgelegt und über das Verteilungsprotokoll werden die Daten an andere bekannte Server repliziert. Diese können dann von den Servern verwendet werden.

Diese Bibliothek wurde in eine bestehende Content Management Plattform integriert. Über eine einfache Benutzerschnittstelle, können die Benutzer die Daten bearbeiten bzw. erstellen.

Als zusätzliche Komponente wurde ein Synchronisierungstool umgesetzt, mit dem es möglich ist, die Daten von einem lokalen Ordner mit einem Server zu synchronisieren. Dieses Tool erkennt automatisch, welche Dateien aktualisiert, erstellt oder gelöscht werden müssen. Diese Funktionalitäten funktionieren bidirektional. Das bedeutet, dass das Tool beidseitig Änderungen erkennen kann.

## Herausforderungen

Während der Entwicklung des Konzeptes stellte sich die Kombination aus Immutable (also unveränderbare) und nicht Immutable Objekten, die im Datenmodell von GIT vorhanden sind, als Herausforderung heraus. Die Datenbank wurde aus diesem Grund während der Entwicklungszeit immer aufwändiger und komplexer.

Für eine Weiterentwicklung des Konzeptes wäre es eine Überlegung wert diese Objekte gesondert zu speichern. Dadurch könnte vieles an Komplexität aus der Datenbank entfernt werden.

Aktuell stellt die Performance des Prototypen und im speziellen des Verteilungsprotokolls die größten Herausforderungen dar. Mit der Implementierung eines "Diff" Algorithmus ein Performancegewinn durch geringeren Datentransfers erreicht werden.

## Fazit & Ausblick

Das in der Arbeit entwickelte Konzept stellt eine gute Grundlage für weitere Entwicklungen dar. Es umfasst alle Facetten von Filesharing und -hosting. Durch die Flexibilität ist es möglich auch große Dateien effizient zu speichern, obwohl sich auch hier Verbesserungsmöglichkeiten während der Projektzeit aufgetan haben.

Eine sehr große Chance für das Konzept wäre es, daraus eine Datenbank zu implementieren, mit der die Plattform und die Clients direkt kommunizieren, um den Overhead der Anwendung so gering wie möglich zu halten.

**Ende**

Danke für Ihre Aufmerksamkeit.